Zoea

Research Note 2

Zoea Concepts and Terminology

Edward McDaid & Sarah McDaid
31 May 2020

This article briefly explains the main concepts and terminology used in Zoea.

Program - Every piece of software that is produced with Zoea is called a program. Every program has a single input and a single output. Inputs and outputs may be a single value like a string or a number, or they may be an array that holds many values. If a program needs to have multiple inputs or multiple outputs then the developer needs to use arrays for these.

Test case - Zoea programs are written by a developer as a set of one or more test cases. A test case is a scenario that describes exactly what should happen when a user runs a program. A simple test case might consist of a specific input (X) and a specific output (Y). This can be interpreted as saying that for this program if the user inputs X then the program will output Y.

Derived value - In addition to input and output a test case can also contain derived values. A derived value is a piece of data that is internal to the program and intermediate in value between the input and the output. Since a derived value is internal it is not something that the end user will see. A test case with an input (A), derived value (B) and output (C) can be interpreted as saying that for this program if the user inputs A then the program will calculate B internally before then calculating and outputting C. Derived values enable Zoea to produce more complicated code by allowing the developer to describe data manipulation as a series of smaller steps.

Step - Each test case consists of any number of steps where a step is an input, an output or a derived value. Each step in a test case has a single associated value. It is possible for a test case to have more than one input steps or more than one output steps. Between input and output steps a test case can also have any number of derive steps. Finally there is no restriction in terms of order for different types of steps. It is valid for a test case to have no steps (does nothing), input only (discards input and does nothing) or output only steps.

Value - A value in a Zoea program can be a number, character, string or a composite value. A composite value can be an array (which is indexed by a number) or a map (which is indexed by a string). Composite values can contain single values or other composite values. As a result values can be any size or complexity.

Element - Every value forms a hierarchy (or tree) made up of one or more nodes. An element is a given node in a value hierarchy. It is like a slot where a value can exist. Every value has a root element. For single value values the root element is the value. For a value that is an array the root element is the array and each value in the array is another element.

Path - Every element has a path that uniquely describes the location of that element relative to the root of the hierarchy. A path is a list of index numbers or keys. The path of the root element is the empty list ([]). For an array the path [1] refers to the first element in that array. For a two dimensional array the path [2,4] refers to the fourth member of the second array. Two different values can have the same element in the sense that they both have a node in their hierarchy with the same path. In this case the values of the two elements can also be different but we still think of them as being the same element.

Relative path - Paths are normally expressed with reference to the root node. We can also define a path relative to any node in the hierarchy. For example the path [2,3,4] relative to root [] is identical to the path [3,4] relative to the node [2]. This is also equivalent to the path [4] relative to the node [2,3].

Data - Depending on the nature of the program there may be a subset of information that is required for every test case. For example if we wanted to describe a program that determines whether an input string is a valid country name we would have to provide a list

of countries along with the input for every test case. This would be inconvenient both for the developer and also for the end user. Instead we can put such information in a special part of the Zoea program called data.

Composition - Larger Zoea programs can be assembled by combining smaller programs. This process is called composition. Zoea does not distinguish between programs, functions or subroutines - these are all just regular programs. A Zoea program can use another existing Zoea program if it has been compiled successfully. The developer can suggest that a program uses one or more existing programs with the 'use' tag. The Zoea compiler may comply with the developer suggestion or it may ignore it and do something else.

Learn more at **zoea.co.uk**